

# **SECURE COMMUNICATION CHANNEL FOR GRID USERS USING AES ENCRYPTION**

A Dissertation Report

Submitted By

**Himansu Shekhar Raman**

To

**Department** of Computer Science & Engineering

In partial fulfilment of the Requirement for the Award of the Degree of

**Master of Technology** in Computer Science and Engineering

**Under the guidance of**

**Mr Aditya Bakshi**

**April 2015**



School of: Technology & Science

DISSERTATION TOPIC APPROVAL PERFORMANCE

Name of the Student: H. S. Raman Registration No: 11307325  
Batch: 2013-2015 Roll No: RK2307A17  
Session: 2014 Parent Section: K2307  
Details of Supervisor: Designation: A-P  
Name: ADITYA BAKSHI Qualification: M.Tech  
U.ID: 17433 Research Experience: 2 years

SPECIALIZATION AREA: Network security of crypto (pick from list of provided specialization areas by DAA)

PROPOSED TOPICS

- 1. Cryptanalysis of AES.
- Application of AES for securing grid computing.
- 3. Improving s-box of AES.

ABakshi  
Signature of Supervisor  
17433

PAC Remarks:

Topic No. 2 is approved

11/01/14  
22/11/14  
Signature: \_\_\_\_\_ Date: \_\_\_\_\_

APPROVAL OF PAC CHAIRPERSON:

- \*Supervisor should finally encircle one topic out of three proposed topics and put up for a approval before Project Approval Committee (PAC)
- \*Original copy of this format after PAC approval will be retained by the student and must be attached in the Project/Dissertation final report.
- \*One copy to be submitted to Supervisor.

## Abstract

TLS protocol provides options to the applications for choosing cipher suite among the set of cipher suites. By making our application to choose one of the strongest cipher suite concluded by various cryptanalysis studies, we can make the application secure. We chose `TLS_RSA_WITH_AES_128_CBC_SHA256`, for our application, grid-user communication so that the communication channel for our request and reply cannot be decrypted easily and also computation for ciphers output can be done in reasonable amount of time. To make the communication secure we should at least use this cipher suite in our application.

## CERTIFICATE

This is to certify that **Himansu Shekhar Raman** has completed M. Tech dissertation titled **Securing Communication Channel for Grid Users Using AES Encryption** under my guidance and supervision. To the best of my knowledge, the present work is the result of his original investigation and study. No part of the dissertation has ever been submitted for any other degree or diploma.

The dissertation is fit for the submission and the partial fulfilment of the conditions for the award of M. Tech Computer Science and Engineering.

Date: First May 2015

Signature of Advisor

(Aditya Bakshi)

## **Acknowledgement**

The Dissertation Secure Channel for Grid Users Using AES Encryption Algorithm is possible with the help and guidance of MR Aditya Bakshi Sir. I am obliged to acknowledge his support.

## **Declaration**

I hereby declare that the dissertation entitled, **Securing Communication Channel for Grid users using AES Encryption** submitted for the M. Tech Degree is entirely my original work and all ideas and references have been duly acknowledged. It does not contain any work for the award of any other degree or diploma.

Date: 1st May 2015

Investigator

Registration No: 11307325

## **Table of Contents**

CHAPTER 1: INTRODUCTION .....	1
CHAPTER 2: REVIEW OF LITREATURE.....	15
CHAPTER 3: PRESENT WORK.....	21
CHAPTER 4: RESULTS AND DISCUSSIONS .....	25
CHAPTER 5: CONCLUSION AND FUTURE SCOPE.....	35
CHAPTER 6: REFERENCES .....	36
CHAPTER 7: APPENDIX .....	37

## LIST OF FIGURES

Figure 1: State matrix .....	2
Figure 2: The overall structure of AES-128 encryption and decryption. ....	4
Figure 3: Encryption round.....	5
Figure4: Shifting of Row. ....	6
Figure 5: Decryption round .....	7
Figure 6: Working of RSA. ....	8
Figure 7: User Privacy Manager .....	22
Figure 8: Flow chart of methodology. ....	24
Figure 9: Client interface. ....	26
Figure 10: Encrypted request.....	27
Figure11: Passing encrypted request to server.....	28
Figure12: Encrypted request received by server.....	29
Figure 13: Registration page for zendesk.....	30
Figure14: Creating company profile.....	30
Figure15: Agent login page.....	31
Figure16: Configuring the server.....	32
Figure17: Customer list page.....	32
Figure18: Unsolved tickets. ....	33
Figure19: Response decrypted at the server.....	34



### 1.1 Introduction to Network and Security

We see computer networks as today's necessity. We deal with small to big computer networks and almost all computer networks are connected to each other in a way or other. We also know that a computer network is as secure as its weakest link is. We require a computer network to be secure against all external attacks and cannot avoid networks be connected to other outside networks.

Therefore we require some tool to save each network from other networks against the possible attacks as well as from inside attacks. One of the best tools to attain security for the network is strong cipher. AES is one such cipher which is able to give us the required security in network communication. The reason that AES can provide us the required security can be well understood by looking at its working principle and various research papers on AES.

#### 1.1.1 Advanced encryption standard (AES):

AES is modern symmetric-key block algorithm that replaced DES and is an open algorithm. AES is an open algorithm which means it is available to public worldwide for use.

#### 1.1.2 History of AES:

NIST started looking for a block cipher replacement for DES in 1997 and announced internationally for responses from all over the world. Cipher would be called as "Advanced Encryption Standard". After second candidate conference of AES which was held in Rome, NIST announced in year 1997 that five out of fifteen candidates were selected as the finalists. After the third AES candidate conference, "Rijndael", designed by Belgian researchers Joan Daemen and Vincent Rijment, was selected as AES in October 2000 by NIST.

AES is one of the best algorithms which are used worldwide.

#### 1.1.3 Working structure of AES:

AES uses substitution-permutation network. The substitution and permutation lets AES for a fast software implementation of the algorithm as was required for the design criteria.

AES encrypts and decrypts a block of 128 bits. Key size for encryption and decryption are 128, 192, or 256 bits dependent on the number of rounds 10, 12 or 14 respectively. All 128 bits of data path in one round gets encrypted or decrypted in AES. Each round consists of four processing steps.

- Byte substitution: Provides confusion.
- Shift row: Provides diffusion.
- Mix column: Provides diffusion.
- Key addition.

Encryption and Decryption uses different order of execution of these four processing steps. Last round doesn't have mix column layer to make the algorithm reversible. At the beginning of AES and at the very end a sub key is added for key whitening.

#### 1.1.4 Internal working of AES:

All operations in AES are done on byte level. The 128 bits data path is split into 16 bytes. These 16 bytes is thought as 4X4 matrix arranged as follows:

$$\begin{bmatrix} H_0 & H_4 & H_8 & H_{12} \\ H_1 & H_5 & H_9 & H_{13} \\ H_2 & H_6 & H_{10} & H_{14} \\ H_3 & H_7 & H_{11} & H_{15} \end{bmatrix}$$

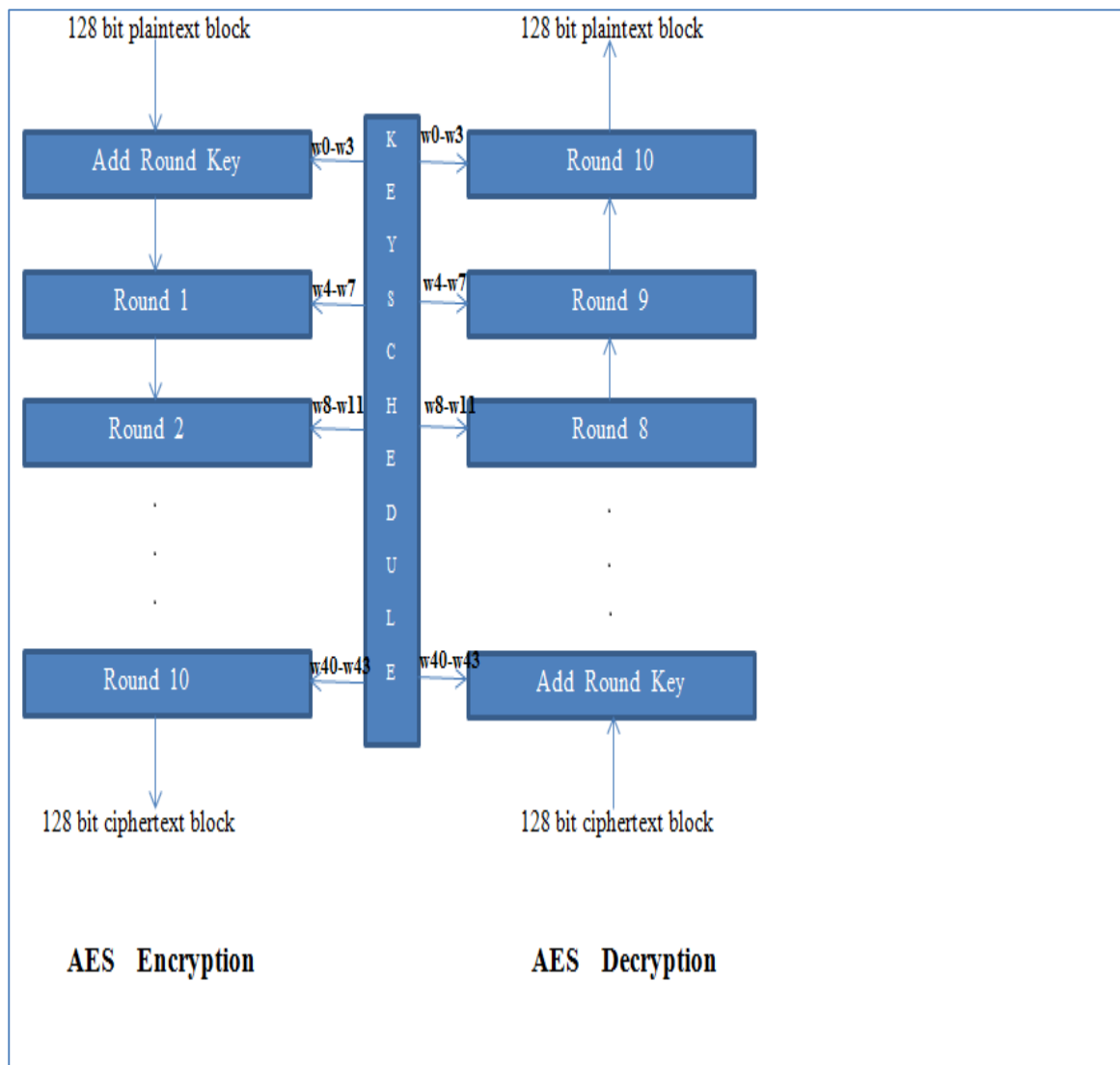
**Figure 1:** State matrix

$[H_0 \ H_4 \ H_8 \ H_{12}]$  or  $\begin{pmatrix} H_0 \\ H_1 \\ H_2 \\ H_3 \end{pmatrix}$  are words. H represents byte here.

128 bit input block is divided into 16 bytes and first four bytes occupies the first column of the matrix, next four second column, next four third column and the last four bytes the fourth column.

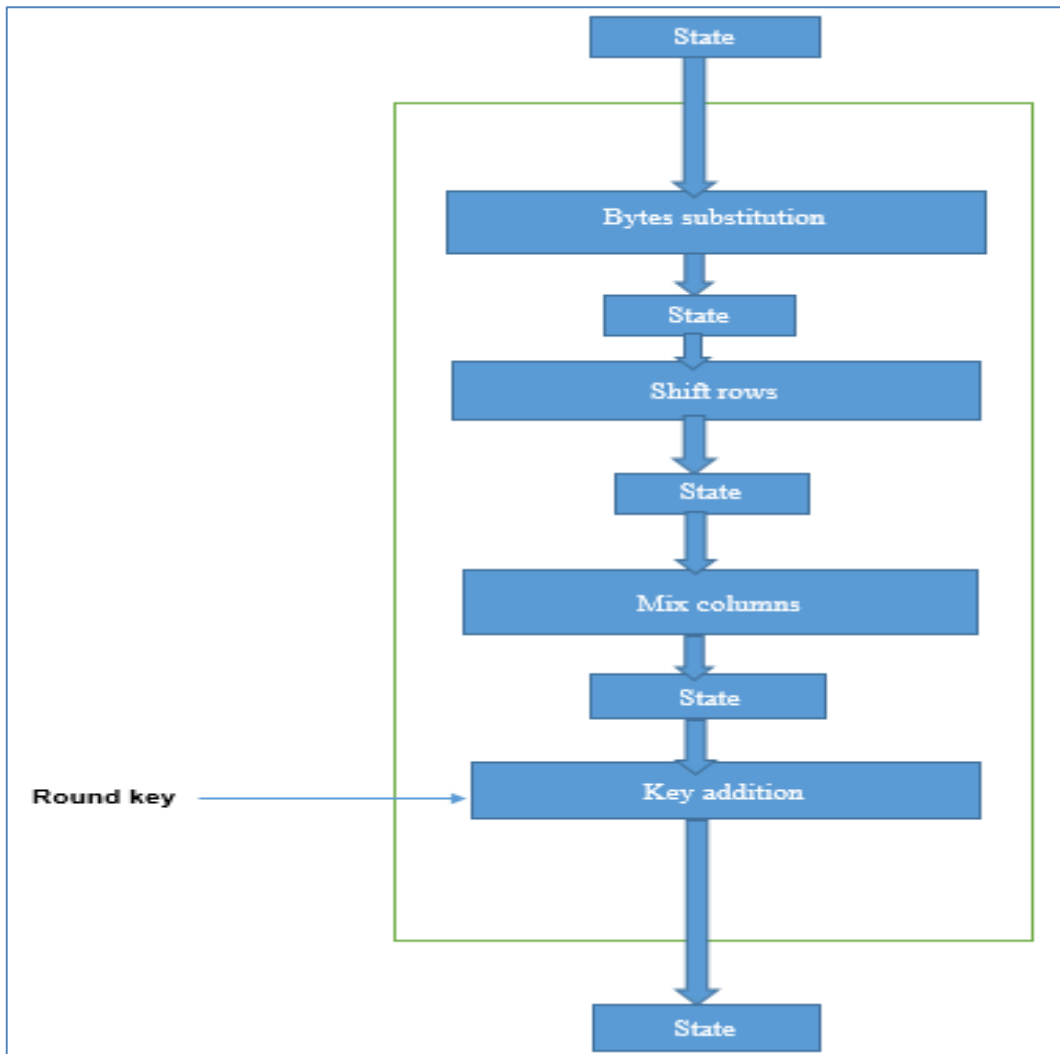
A word is four bytes in AES. So, we say each column or each row of state matrix a word. Input state matrix is converted into output state matrix and each round works on state matrix. The last round output matrix is arranged as output block. Encryption and decryption algorithms of AES are not carried out exactly in the same ways; they are different in order of execution of steps.

We take AES-128 with 128-bit key which is arranged into  $4 * 4$  bytes matrix. When the input blocks comes, first word from key seals first columns of the matrix. The four words or column of key matrix are expanded into 44 words. Each round of algorithm ingests four words from the key agenda.



**Figure 2:** The overall structure of AES-128 encryption and decryption.

Each round has four processing steps: The following diagram shows the processing of each round except the last round.



**Figure 3:** Encryption round.

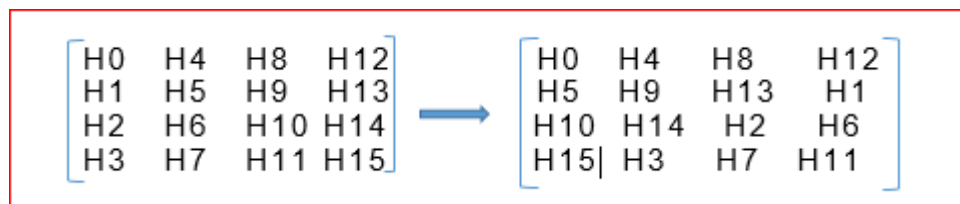
**Bytes substitution:**

The purpose of this step is to reduce the association between the input bits and the output bits. It creates confusion for the algorithm. Substitution is done byte by byte using the lookup table of 16X16 size, called as, “S-Box”. Each cell of table is filled up by Galois field arithmetic  $GF(2^8)$  and bit scrambling. The bit scrambling sub step makes sure that the substitution cannot be defined in the form of any easy mathematical function. We group the given input byte into two 4-bit subgroups, each giving an integer value and represent them by their hexadecimal equivalent. Then we use the left 4-bit value to look the row and right 4-bit value to look for column of the table value and then the value at the intersection is substituted as a byte for shift row step. The hexadecimal value of zero of input byte is substituted by zero.

During Decryption process the construction of S-BOX is reversed then, we did while constructing s-box during encryption.

Byte substitution should be invertible, for that we need one to one mapping for substitution of bytes. So that for each input byte there should be unique output byte and vice versa. No input byte should be replaced by itself except zero.

The shift rows step: In shift rows the state matrix is shifted row wise as: first row no shift, second row shifts left circular by one byte, next row shifts left circular by two bytes and fourth row by three bytes left as:



**Figure 4:** Shifting of Row.

The advantage of doing this is scrambling of byte order of the input block, which in turn creates diffusion. Scrambling occurs because, while forming state matrix we arrange bytes column wise and this circular left shift scrambles the original byte order of input block.

While decryption the shifting is circular right of bytes row wise. First row not changed, second row by one byte, next row by two bytes and fourth by three bytes.

**Mix columns:**

This step mixes bytes in each column. During decryption this step is called as inverse mix column. Each byte in a column is replaced by performing arithmetic on elements of the same column as described by the equation:

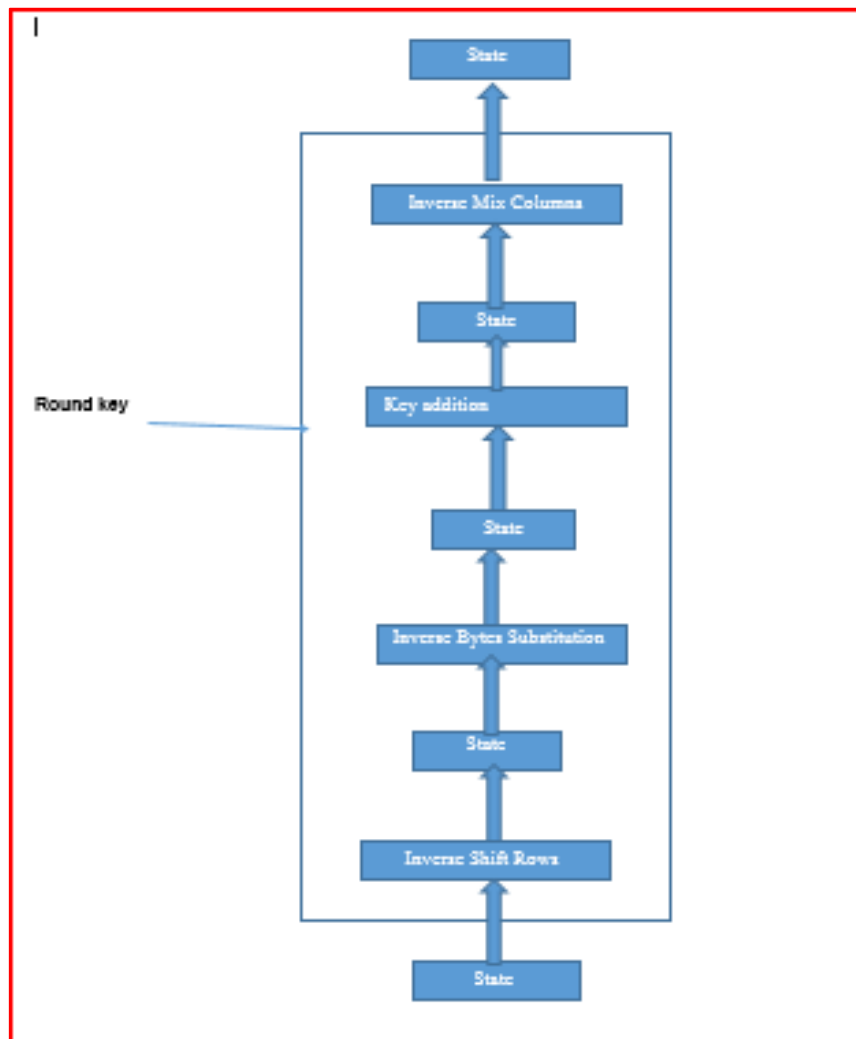
$$E'(1,j) = (0x02 * E(1,j)) + (0x03 * E(2,j)) + E(3,j) + E(4,j)$$

Where rows and columns are numbered as 1,2,3,4.

Value of j = 1, 2, 3, 4

The addition and multiplications performed are according to GF (2<sup>8</sup>). Arithmetic addition is simply XOR operations. The operation on first row elements can be described as:

Add round key: During encryption this step adds the round key to the output of previous processed step. During decryption this step is known as inverse add round key.



**Figure 5:** Decryption round

### 1.2. RSA cipher:

This public key cipher is in practise for symmetric key exchange by encrypting them and also for digital signature, because of its robustness and security provided by it. It

is relatively slower in processing therefore we do not use it for large amount of data encryption. Symmetric key ciphers requires, secret key for encrypting and decrypting, we need secure method to share secret key. One of the methods for sharing secret key is using public key cipher and RSA is one of them. This cipher is very useful for the purpose of key exchange. We can use it for small data encryption or decryption, for example, symmetric key. It does combine server authentication with key exchange. Server while sending its certificate to the client also sends the public key within the certificate itself. Private key of the server should be changed regularly to avoid attacks.

This is a public key cipher which uses modular exponentiation. First step of this algorithm is key generation than encryption and Decryption takes place using the generated keys. It generates two keys which are public and private. Private keys are always kept secret. Data encrypted by one key is decrypted by the other key. Depending upon our requirement we decide which key should be used for encryption and which key should be used for decryption.

**Key generation:**

1. Choose two large prime numbers: P and Q of similar bit length, randomly.
2. We calculate  $N = P * Q$ .
3. We calculate  $\phi(N) = \phi(P) * \phi(Q) = (P-1) * (Q-1) = \{N - (P+Q-1)\}$
4. We choose an integer "E" such that:  
 $1 < E < \phi(N)$  and  $GCD(E, \phi(N)) = 1$
5. Calculate  $D = E$  inverse (mod  $(\phi(N))$ ).
6. Public key = (N, E).
7. Private key = (N, D).]
8. P, Q, D and  $\phi(N)$  are kept secret.

**Encryption:** Plaintext = M < N.  
Cipher text.  $M^E \pmod{N}$ .

**Decryption:** Cipher text = C.  
Plaintext:  $M = C^D \pmod{N}$ .

**Figure 6:** Working of RSA.



### **1.3 We need to achieve some security goals for server system as:**

- Every user or computer must be identified before they can access to the resources. For this we need to collect user information and store them.
- User authentication must be done. We should ask for user name, password and other information related to the user. We can then allow users to access resources for which they are eligible.
- We should be able to identify and stop the modification of data while the data is passing through the communication channel.
- Users should not be able to deny the fact that they have committed something which they already have done.

### **1.4 Transport layer security (TLS):**

It is designed to protect privacy and integrity of data between communicating client and server. It has two sub layers, record protocol and handshake protocol.

It is not dependent on protocols which are above the transport layer. It is our responsibility to decide that how we want to make TLS handshaking and authenticate certificates as designer of higher level protocols than transport layer. Cryptographic analysis done by us, helps in designing protocols and using ciphers which secures our applications. We as a designer has to decide on various attributes like digital signature, block cipher encryption, and public key encryption etc.

#### **1.4.1 Record protocol:**

It is concerned with making connection private using symmetric cipher by making available unique key for every connection. We can use record protocol without encryption also. It also makes connection reliable by implementing integrity check for the messages using MAC, such as SHA-2. It uses keyed MAC to protect message integrity. TLS recommends us to use SHA-256 or more robust standard hash function than this. We need to design any protocol cautiously so that we can prevent any possible attack. We should depend on our cryptanalytic analysis rather than on to TLS.

This protocol is itself layered and each layer has some function. It receives message from upper layers and fragment them into useful blocks, compresses the data if required, applies MAC, encrypts and transmit the result to the receiver of the message. After receiving the data from sender it decrypts, verifies, decompresses, reassembles and then delivered to the

higher layer. This layer receives data from its upper layers and performs the following actions.

- **Fragmentation:** This sub layer fragments message into size of  $2^{14}$  bytes or less.
- **Compression and Decompression:** Message in this layer is compressed or decompressed according to the active algorithm in the session. It must be lossless.
- It evaluates MAC then encrypt the data and the MAC also.
- **Padding:** To apply block cipher to the data received from upper layers or after processing done by record layer itself, we need to add bytes of data to the message such that size of the block becomes integral multiple of block size. It can add up to 255 bytes.

#### **1.4.2 Handshake protocol:**

It works above the record layer protocol. It decides on tools to be used for cryptography. It is used for authenticating client and server with one another. It also decides the cipher to be used and cryptographic keys before the communication begin and before the data transfer actually starts between them. It makes connection secure with the help of authentication (client and/or server) using RSA, DSA; making shared secret secure so that no one trying to access communication channel can get to know about the shared secret key and if done can be detected easily. It uses three sub protocols. The layers above it should decide on which set of algorithm can be used so that communication will be secure against any attack. During design of implementation of our application, we should take care of things like which signature and hash algorithms can be used together, because TLS does not support every possible combination.

##### **1.4.2.1 Session negotiation:**

During negotiation of session various things require like session identifier which is random byte sequence generated by server to identify session state. Certification of client and server is done, compression technique for compression of data before encryption. Termination of session is done at the end and can be started by either client or sever.

#### **1.4.2.2 Cipher suite definitions:**

It decides on which cipher suite to be used during the session between client and server. One valid example is TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256. It means that TLS is using RSA, AES and SHA.

#### **1.5 Importance of Security Requirement in Grid Computing which is distributed:**

It is the collection of the various computational resources, from multiple administrative domains and reaches a common goal, transparently from the users (who requested for the service). Distributed computing makes possible solve the computational tasks which require massive computational power which will not able to be handled by single resource or processing element.

Since multiple domains, various resources are involved we require security to secure the system from unauthorized access. We are implementing security in communication channel so that no user will be able to see other users' requirement and hence cannot be misused by any particular user.

#### **1.6 Microsoft Visual Studio:**

It is an IDE which we use for developing software; it has a code editor, a source-level debugger and machine level debugger. It also has tools like form designer which we use for developing GUI applications, we use it for designing and developing websites, class and database schema also. It supports many programming languages. It has some built in languages and also supports some languages which can be installed by using language services. Native codes are machine codes which as we know executed by CPU directly and managed code (source code) both can be produced by it.

- There are many integrated development environments are available in the software development environment. In of the visual studio is one of the integrated development environments.
- The main functioning of the visual studio is to develop various types of the applications that may be windows or web console applications.
- Windows console applications are the applications that are used for the internal purpose of a single distributed system such as a network server connected to many systems.

- Web console application are the development of applications for a particular organization or any other that may acts as a customer for their own development of the application.
  - The main features of Visual Studio:
    - Code Editor
    - Debugger
    - Designer
    - Other Tools
- Code Editor is the important feature that is used for writing the code that which it has in built syntax and easy to access them at any time by just typing the name of the class.
- Debugger is available in visual studio that acts as both the machine and source level debugger. It also simply manages with the code and the core code which is written by own for the program of the application of any language the program may be written.
- Designer is the one use of designs for the helpful in developing the applications. It contains various types of designers that are used for the development of the application. Designs can be of various types and the designs are selected according to the developer's issue. Various types of designers
  - Windows Form designer
  - Class Designer
  - WPF Designer
  - Mapping Designer.
  - Data Designer
  - Web Designer or development
- Other tools include open tab browser, Team explorer, Data explorer, Server explorer, properties editor, object browser, solution explorer, dotfuscator software services community edition, ASP.NET website management tool, Text generation framework, Visual studio tools for office.

## 1.7 .NET Frame Work:

- A frame of software which was developed by Microsoft that works on the platform that is developed by Microsoft. It consists of various types of classes and libraries of various functionalities that are available in other type of programming languages.
- The main function of this frame work is to get the access of the various functionalities of other ones and so that it is easy to use the functions that are used for the development of any application or the other console application.
- Two types of application are developed through this frame work they are Windows Application and Console or Web Application.
- There is a super class that contains various classes called as framework class library. FCL (Framework Class library) consists network procedure communication, algorithms of numeric, encryption of the code, data accessing activities like data abstraction, connectivity to the database and various functionalities.
- Programs can be written very easily in .NET Framework as it consists of all types of classes and functions which are in built in the FCL (Framework Class Library) so that it save more time and also in economical way.
- .NET framework architecture consists of the .NET core, Assemblies, class library and common language infrastructure.
- Common language infrastructure (CLI) acts as a platform to obtain the needful functionalities for the execution and development of the application in a simple way that the other frameworks that are available.
- Class Library consists of all types of classes that are collected through the FCL classes for file handling, exception handling and many others. Standard libraries of classes are available so that they are ordered name space of the hierarchies.
- Base class library is a part of the class library and it also acts as a super class it contains all the information about the classes that are required for developing the application. Various functions such as LNQ, WCF, WF, ASP.NET and ADO.NET are include in the base class collections.
- Framework Class Library is a small part of the class library and it contains .dll files which are available only in the Framework class library that are only for the development of the application or the windows form

- .NET core is one of the implementation prototype of .NET Framework which is incomplete that means cannot function the total implementation of the .NET Framework.
- Assemblies are used for the storage of the code that are present in the CLI. These are called simply CLI assemblies. These CLI assemblies are used for the .dll and .exe files that are obtained by the developing the application using the FCL and CLI of the .NET Framework.
- The development tool that is used by the .NET Framework is Microsoft Visual Studio, operating system is windows, server is oracle access and many other functions are used by the .NET Framework.

## Chapter 2

### Review of Literature

---

Paper on AES cryptanalysis says that all the cryptanalysis techniques present today needs improvements before they can be able to cause any harm to AES. The good thing is that there are three different types of AES that we can use depending upon our requirement of security goal. Various cryptanalysis techniques for AES are studied [1] and the study shows the following:

#### **2.1 Attacks before creation of AES:**

Attacks that were present at the time of creation of AES were linear, differential, boomerang attack and others. During the creation of AES the designers of the algorithm applied constraints so that there are neither linear trails nor differential trails present. AES fulfils the conditions for the algorithm to be resistant against such attacks for 8 rounds and more. Hence secure against linear and differential cryptanalysis. AES is also secure against 7 and more rounds for square attack and truncated differentials. AES is also secure against interpolation attacks.

#### **2.2 The attacks after creation of AES are:**

##### **2.2.1 Algebraic attacks:-**

This attack is designed to find the key of AES. Most of the algebraic attacks do not requires very large number of plaintext, cipher text pairs when key is unknown. Classical linear attacks on DES requires  $2^{40}$  such pairs. But there is no substantial proof that these algebraic calculations are able to break AES.

Extended linearization developed by Courtios, [2] was line of attack aimed specially for usable to attack AES. Improvement of extended linearization was also developed by Courtios and Pieprzyk. There are many papers and available to further improvements related to this approach. Studies on similar approaches results as unworkable, as intended.

##### **2.2.2 Cube attacks:**

This attack is no better than brute force attack for searching key space practically against AES [3]. AES was not overdesigned against algebraic attacks because these attacks were came into existence after the design of AES itself but AES is still immune to such attacks till now. AES as we know is designed using algebra and is a very elegant algorithm and it is obvious to think that algebraic attacks can break it but much progress is required in the field of algebra to break AES.

### **2.2.3 SAT solver and Hybrid attacks:**

- Boolean satisfiability problem.
- Hybrid attacks analysis.

#### **Boolean satisfiability problem**

We can represent block ciphers as Boolean expression. The variables of the expression are plaintext; key input bits and output bits. Plaintext bits when encrypted into cipher text bits with the key bits then only the constructed Boolean expression is true. Block cipher is attacked by setting plaintext and cipher text into Boolean expression. We then solve the Boolean expression and try to find key variables which make expression correct.

SAT solver: It is a software which provides the answer to satisfiability problem automatically. E.g. z chaff, MiniSat, SAT4J are open-source Sat solvers.

It cautiously gives values to the variables one by one till contradiction is met and the Boolean expression turn out to be false. Sat-solver then goes back and starts again and again inserting other values for variables to escape contradiction. This contradiction-focused going back again and again reduces large part of the search space. Hence the SAT solver is efficient than brute force search in terms of consuming time.

This approach found key for two and three rounds of DES. But theoretically it was supposed that SAT solvers can find the key for all form of AES. SAT solvers are computationally not efficient because it consumes much time in finding the solution. Boolean formula contradiction for DES requires majority of unknown key variables checked. It takes almost the same time as a brute force search hence, AES cannot be attacked effectively by such programs.

### **2.2.4 Hybrid attacks analysis:**

SAT solver can be of great use when we combine it with other techniques [4]. Study on DES, triple DES and AES has been done by combining side-channel and SAT-solver attack. Conclusions of the study is – if a side-channel attack is done and we can obtain values of input and output bits of any one of the ten rounds of AES, the SAT solver will be able to find the full key. But the investigators did not bother to find all the inputs and outputs of a round using side-channel technique. Hence, it cannot be said that this attack is feasible until work is done on side-channel attack.

### **2.2.5 Side Channel Attacks:**



This attack make use of information gathered from a cryptosystem due to exposures in its physical implementation instead of any cryptographic vulnerabilities of the algorithm.

### **2.2.6 Timing attacks:**

The most capable developments of this attack is based on implementation of AES that is, “micro-architectural” properties of the hardware.

### **2.2.7 Power analysis:**

Power used by cache usage and other micro-architectural mechanisms can be studied and used to analyse secret key information. Power analysis can be done remotely or by covertly inserting a monitoring device in the target device. This attack requires very small number of samples, which helps in eluding intricate statistical method. 128 bit key was obtained by measuring 40 power samples only [5].

### **2.2.8 Fault injection analysis:**

AES has been found to be fault sensitive which means AES is sensitive to fault analysis. For this attack to work we require device in which we can reliably induce faults. Induction of faults should be induced in such a way so that device does not permanently get damaged. The attacker must be in control of the encrypting system, physically, to execute this attack [6].

This attack can be stopped by deactivating device automatically when some finite number of faults observed, so that enough information can't be collected for attack to be effective. In practical implementation of Digital signature using RSA, the main problem is the lot of time required to generate keys and the hardware required, especially large memory. Time required to generate the key requires multiplication of two large prime numbers of size 1024 bits or more for security reasons. Since the compilers that we do have, operates normally on 64 bits, we need some method or function that can compute product of such large numbers (1024 bits). The paper, [7], is one such example. In this paper authors tried to use complex numeric function to ease the process. Using the process it is possible to create 1024 bits key in less than two minutes on common computer system. Data could be encrypted or decrypted in less than two seconds. So efficiency of RSA is better than before and thus we can use RSA with longer keys and will be more secure and fast at the same time on any general computer system.

## **2.3 Francois-Xavier Standaert: “*Evaluation Tools for Side-Channel Attacks*”.**

In this paper they discussed about the side channel attacks. Firstly define in side channel attack i.e Context with respect of measurement that describe the leakage distribution of target device, then device input output control can be chosen adversary .lastly to check the complexity based upon data, time, memory and number of measurements. Implementation can be described mainly on 2 aspects. Firstly Design type, in this how implementation criteria taken into consideration i.e based on randomization and also with time randomization secondly leakage type function provide features like are linearity, noise distribution and variability. Lastly conclusion of this paper is to evaluate the attacks and the method that was adopted for the cryptanalysis.

#### **2.4 Amir Moradi, Markus Kasper, Christof Paar. “*On the Portability of Side-Channel Attacks*”.**

In this paper they are providing summary about real world side channel analysis of encryption mechanism. They are using Xilinx FPGAs encryption mechanism. For analysis virtex 4,virtex 5,and Spartan 6 is used to show that encryption mechanism can be completely broken by applying some methods like correlation power analysis that is used to measure the power consumption of a device when the algorithm execute. Here results shows the measurement of the algorithm based on the power consumption. For virtex 4 the power consumption measured is 50000 and for virtex 5, 90000. Analysis is done to find time instances of power traces with the help of known key where decryption happens. In today's world important topic is of IP theft and piracy of the product that results insecurity. The attacks which is happened beyond 8 bit hypothesis that were taken into consideration for implementing a more secure system.

#### **2.5 ANDREW W. APPEL, Princeton University “*Verification of a Cryptographic Primitive: SHA-256*”:**

Cryptography is tough to try and do right, and also the solely thanks to recognize if one thing was done right is to be able to examine it. This argues terribly powerfully for open supply cryptographical algorithms. But merely commercial enterprise the code doesn't mechanically mean that individuals can examine it for security flaws. Bruce Schneier be suspicious of economic encoding computer code, because of back doors attempt to use public-domain encoding that must be compatible with other implementations. Bruce Schneier in later years use wide used, well examined ASCII text file implementations of printed, nonproprietary, wide used, well examined, commonplace algorithms—because

“many eyes create all bugs shallow” works provided that there are several eyes taking note. To this we add: use implementations that are formally verified with machine-checked proofs of practical correctness, of side-channel resistance, of information-flow properties.

Generally it takes some of years to notice the bugs. Verification will guarantee program properties prior to of widespread unleash. In this paper there is a primary step: a proper verification of the practical correctness of the SHA-256 implementation from the OpenSSL ASCII text file distribution. Formal verification isn't essentially a substitute for many-eyes assurance. For instance, in this case, I gift solely the reassurance of practical correctness (and its corollary, safety, as well as absence of buffer overruns). With regard to alternative properties such as temporal arrangement facet channels, it prove nothing; thus it's comforting that this same C program has over a decade of widespread use and examination. SHA-256, the Secure Hash algorithmic program with 256-bit digests, isn't an encoding algorithmic program, but it's employed in encoding protocols. The ways they discussed during this paper can be applied to constant problems that seem in ciphers like AES: interpretation of standards documents, big-endian protocols enforced on little-endian machines, odd corners of the C linguistics, storing bytes and loading words, signed and unsigned arithmetic, extended preciseness arithmetic, trait of C compilers, use of machine-dependent special directions to create things quicker, correspondence of models to programs, assessing the trusty base of the verification tools.

This paper presents the subsequent result: they actually have proven useful correctness of the OpenSSL implementation of SHA-256, with relevancy a useful specification: a formalization of the FIPS 180-4 Secure Hash normal [FIPS 2012]. The machine checked proof is completed mistreatment the Verifiable program logic, within the proof assistant. Verifiable C is proven sound with relevancy the operational linguistics of C. The program may be compiled to x86 programming language with the optimizing C compiler; that compiler is proven correct with relevancy an equivalent operational linguistics of C and also the linguistics of x86 programming language. Thus, by composition of machine-checked proofs with no gaps, the assembly-language program properly implements the useful specification. In addition, they enforced SHA-256 as a useful program and proven it like the useful specification. It will execute the useful program on real strings (only 1,000,000 times slower than the C program), and gets an equivalent answer as normal reference implementations. This offers some further confidence that no silly things are wrong with the useful verbal description.

**Limitations:**

The implementation is from OpenSSL, with some macro growth to instantiate it from generic SHA-2 to SHA-256. CompCert generates programming language, not machine language; there's no correctness proof of the program or of the x86 processor hardware on that one may run the compiled program. The proof assistant is wide used, and its kernel is believed to be an accurate implementation of the predicative Calculus of Inductive Constructions, that successively is believed to be consistent. A different reasonable limitation is within the time and price of doing the verification. SHA-256 was the “shakedown cruise” for the Verifiable C system. This “cruise” disclosed many inefficiencies of Verifiable C’s proof automation system: it's slow, it's a memory hog and it's troublesome to use in places, and it's incomplete: some corners of the C language have inadequate automation support. However, this integrity, or shakiness of proof automation, cannot compromise the end-to-end guarantee of machine-checked logical correctness: each proof step is checked.

## Chapter 3

### Present Work

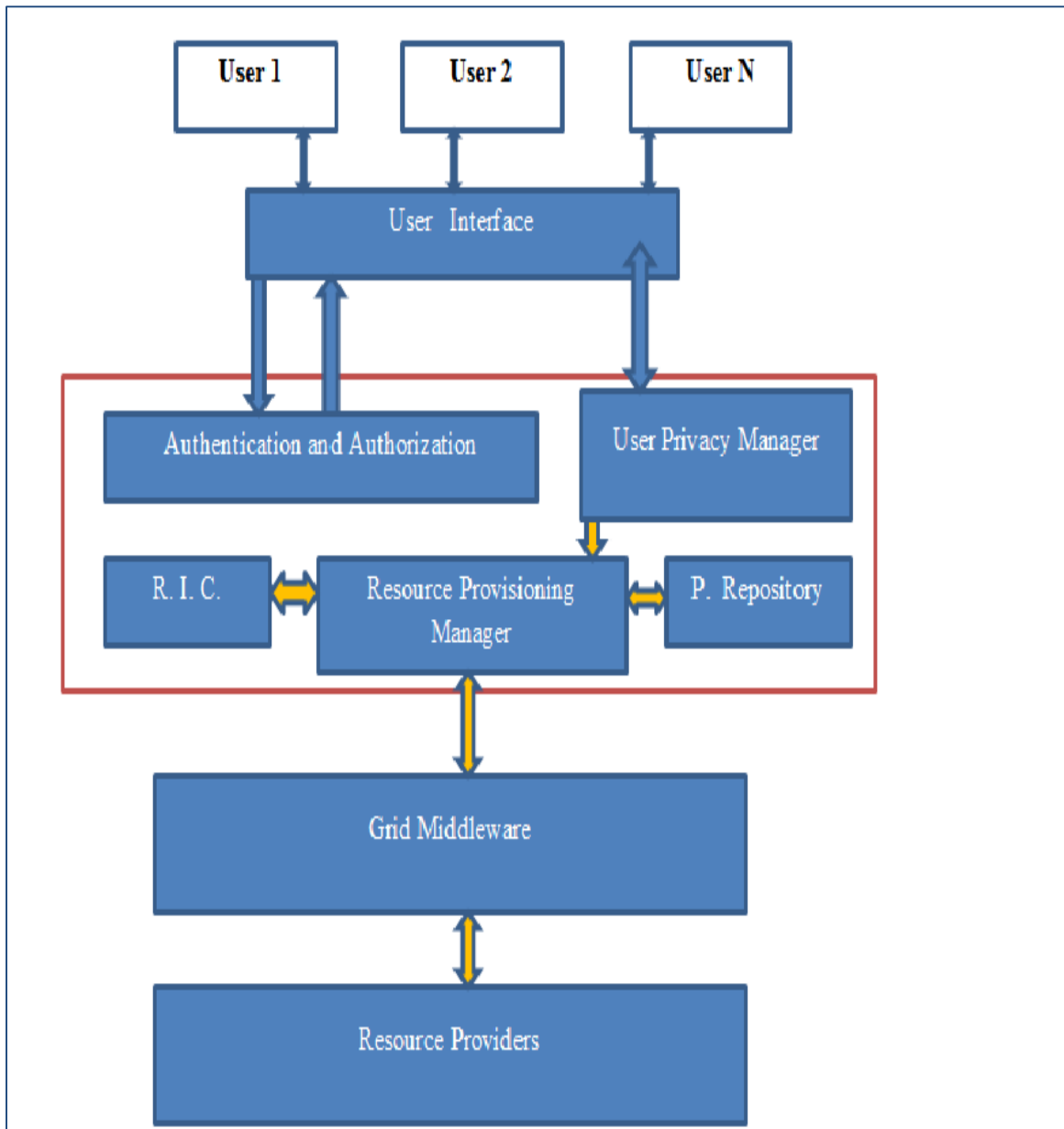
---

#### **3.1 PROBLEM FORMULATION:**

We know that Grid computing is very useful for resource services and it also provides us with great computing power as well as resource sharing. Most of the work in the field of Grid security discusses about core security issues which are very interesting and important. But we also need to secure the communication channel, because if it is not secure than it will become the weak link where attack could be triggered. Also if we do not apply security in channel communication, usefulness will get compromised because input and output could be intercepted and analysed by others. So to assure security in communication channel we are applying strong cipher set which will help to improve robustness and reliability of Grid computing.

#### **3.2 OBJECTIVE:**

Our work is to design User Privacy Manager such that every user who will use services of grid computing will be secure against each other and from outside attacks as well. So our objective is to improve security aspect of QoS for Grid environment and justifying the process. Our aim is to secure communication channel used for the distributed environment which is very important field of research. Our proposed model will secure channel for grid users using AES encryption algorithm for.



**Figure 7:** User Privacy Manager

### **3.3 METHODOLOGY:**

Client during TLS handshake will tell server that which set of protocols it is using. Security provider will check if client is using the predefined set of protocols or not. If client is using the same set of protocols as that of server than only the following steps will occur using those set of protocols.

**Step one:** Client and server will ask each other for authentication.

**Step two:** Client will ask for communication with server using AES, RSA and SHA.

**Step three:** Server will agree for communication using AES, RSA and SHA.

**Step four:** RSA key exchange method will be used for exchanging the key of AES.

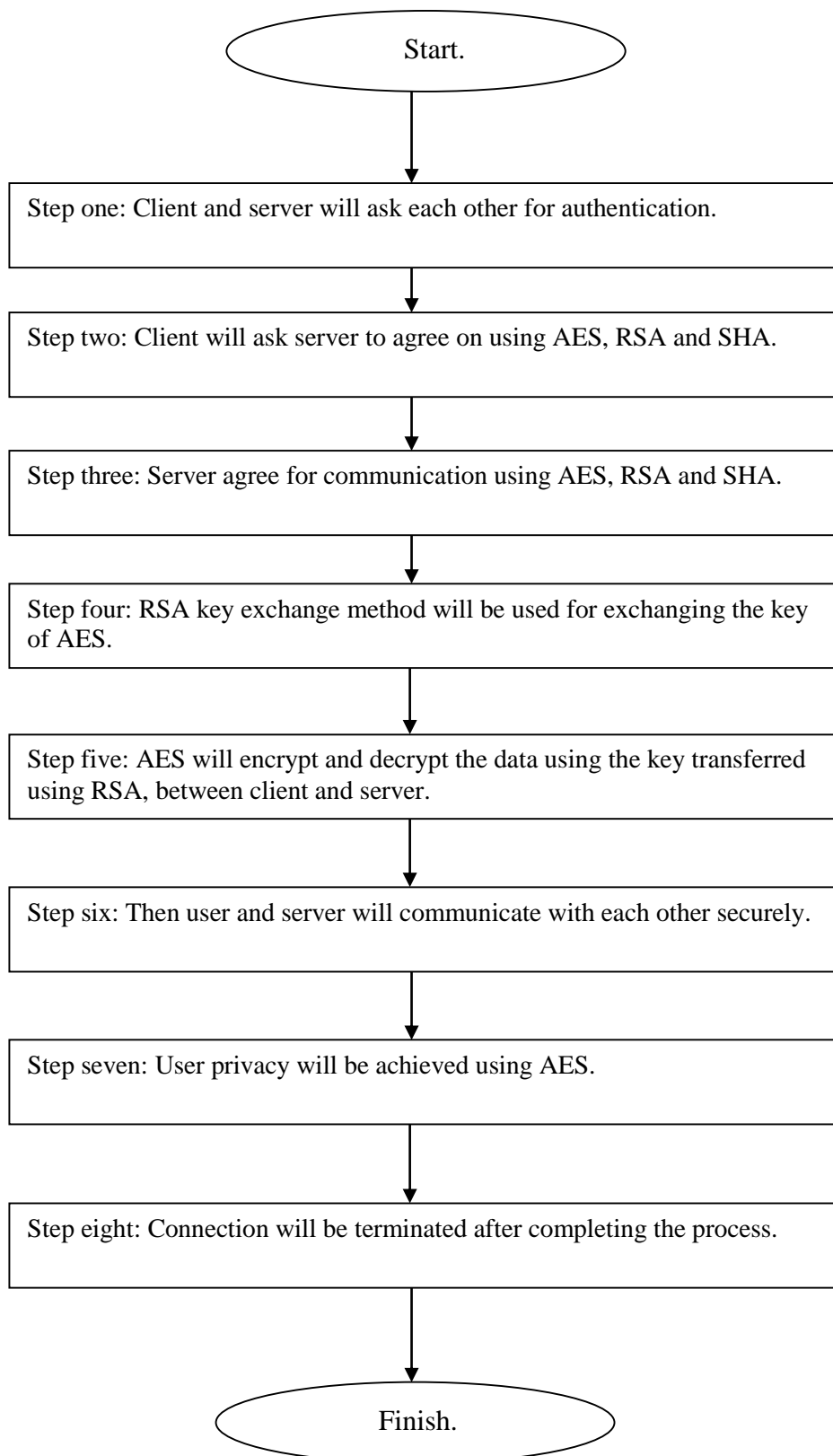
**Step five:** AES will encrypt and decrypt the data using the key transferred using RSA, between client and server.

**Step six:** Then user and server will communicate with each other securely.

**Step seven:** User privacy will be achieved using AES.

**Step eight:** Connection will be terminated after completing the process.

Only the authorized client will be able to use the services of the system since authentication process is done at the first place. Only the certified server will be able to service the authenticated client. All users will be secure with each other and no user will be able to know others data or message. Hence privacy is maintained. We will attain privacy in communication using AES algorithm which is one of the most secure algorithm.



**Figure 8:** Flow chart of methodology.



# RESULTS AND DISCUSSIONS

---

### RESULTS:

Outcome of proposed system, “Secure channel for grid users using AES encryption algorithm”, provides secure communication for users in grid environment and in distributed environment. It will be difficult for anyone to see others data.

### DISCUSSIONS:

Scientific, research and even general applications and various user requirements in future will require more resources than we need them now. So we will be in need of using grid computing services more often. Isolated grid system will not be able to utilize the resources economically. We need commercialization of grid systems which means we are going to provide grid services on Internet infrastructure as well. We can charge accordingly for providing grid services. Client and providers both will get benefits by doing this. Providing grid services on Internet infrastructure is useful but then we will have to be cautious about its security aspect.

The weakest link could be the security of communication channel, which is public. Various protocols like TLS are there to ensure security but attack on them is possible if we do not strictly design our system to use strongest set of ciphers. Being strict on choosing strongest set of ciphers has advantage of secure system. But doing this will have disadvantages also, like users using old software for example old Operating systems, old web browsers etc. will not be able to access the services. This problem of unavailability has very simple solution, which is updating the software regularly.

## Passing Request To server

Subject: 

Priority

 normal high

## Viewing The Requests are encrypted

subject	priority	status	type
CgATB68taOE+DW2iup0gHg==	normal	open	
G/XbUYvvNm4=	normal	open	
0yyFxdyOF7vagz0t3ANev==	normal	open	
yS7wI7R1I2HIs0ioOfopQ==	normal	open	
GSqs/bcOIL3tb3aj5iVhiv==	normal	open	
BroLvraFzYs=	normal	open	
mXMCGRHMK1UathFPYSbuw==	normal	open	
rQzo8QQ4Jl=	normal	open	

Internet Download Manager

**Figure 9:** Client interface.

## Viewing The Requests are encrypted

subject	priority	status	type
CgATB68taOE+DW2iup0gHg==	normal	open	
G/xbUYvvNm4=	normal	open	
0yyFxdyOF7vagz0t3ANew==	normal	open	
yS7w7R1I2Hls0ioOFoapQ==	normal	open	
GSqs/bcOIL3tb3aj5Vhiw==	normal	open	
BroLwfaFzYs=	normal	open	
mXMCGrHMK1UathFPYSbuw==	normal	open	
rQzq8QQ4J4l=	normal	open	
ody5rB/0wu0=	normal	open	
19y14jrCuL1UathFPYSbuw==	normal	open	
19y14jrCuL1UathFPYSbuw==	normal	open	
GRHW7ClebP/MaYBTOhog8w==	normal	open	
khg		open	

## The Response are Decrypted


[Response](#)

**Figure 10:** Encrypted request.

## Passing Request To server

Subject: 

Priority

 normal high[Request-server](#)  for send encrypted request

## Viewing The Requests are encrypted

subject	priority	status	type
CgATB60taOE+DW2iup0gHg==	normal	open	
G/XbUYvvNm4=	normal	open	
0yyFxdyOF7vagz0t3ANew==	normal	open	
yS7wI7R1I2HIs0ioOfogpQ==	normal	open	
GSqs/bcOIL3tb3aj5Ivhiw==	normal	open	
BroLwfaFzYs=	normal	open	
mXMCGrRHmk1UathFPYSbuw==	normal	open	

Internet Download Manager

**Figure 11:** Passing encrypted request to server.

## Request-server

## Viewing The Requests are encrypted

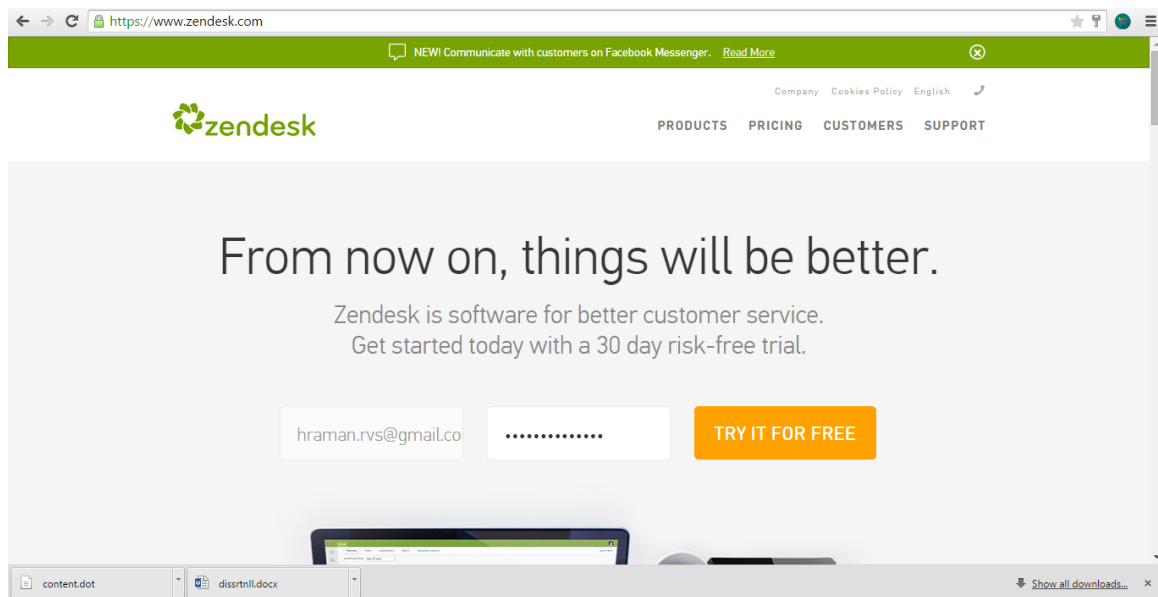
subject	priority	status	type
CgATB68taOE+DW2iup0gHg==	normal	open	
G/XbUYwNm4=	normal	open	
0yyFxxgdyOF7vagz0t3ANew==	normal	open	
yS7w7R112Hls0io0FogpQ==	normal	open	
GSqs/bcOIL3tb3aj5iVhiw==	normal	open	
BroLvifFzYs=	normal	open	
mXMCGrRHmk1UathFPYSbuw==	normal	open	
rQzq8QQ4J4I=	normal	open	
ody5tB/0wu0=	normal	open	
19y14jrCuL1UathFPYSbuw==	normal	open	
19y14jrCuL1UathFPYSbuw==	normal	open	
GRHW7ClebP/MaYBTOhog8v==	normal	open	
khg		open	
0WfhOClm4/3Ug5dxaotmw==	normal	open	

 last request ...

**Figure 12:** Encrypted request received by server.

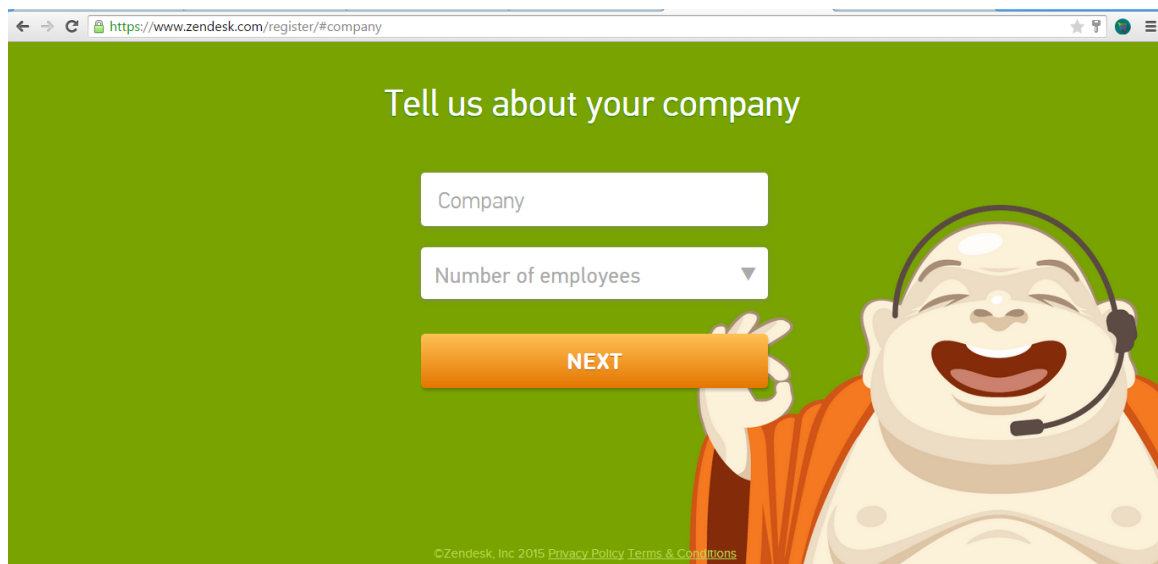
#### 4.1 About zendesk:

It provides various services and we are using it as our distributed server.



**Figure 13:** Registration page for zendesk.

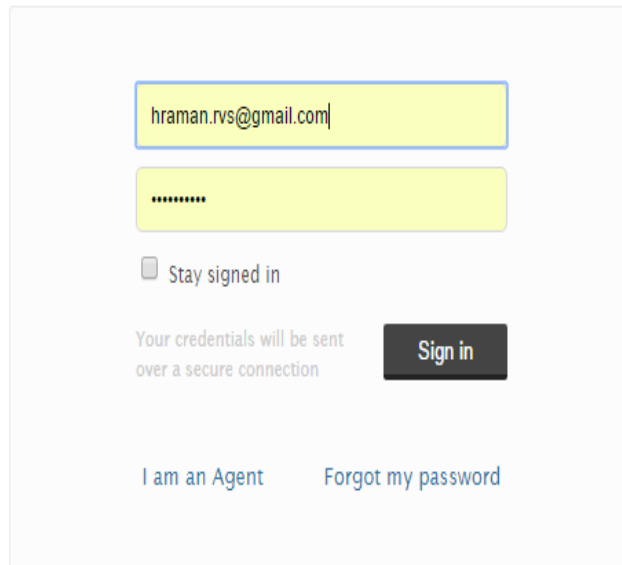
We are using trial version of zendesk which is provided to us for 30 days after registering, we can use its services for free.



**Figure 14:** Creating company profile

We are using it as a third party service provider. We are using it for distributed computing server.

## Sign in to secingrid

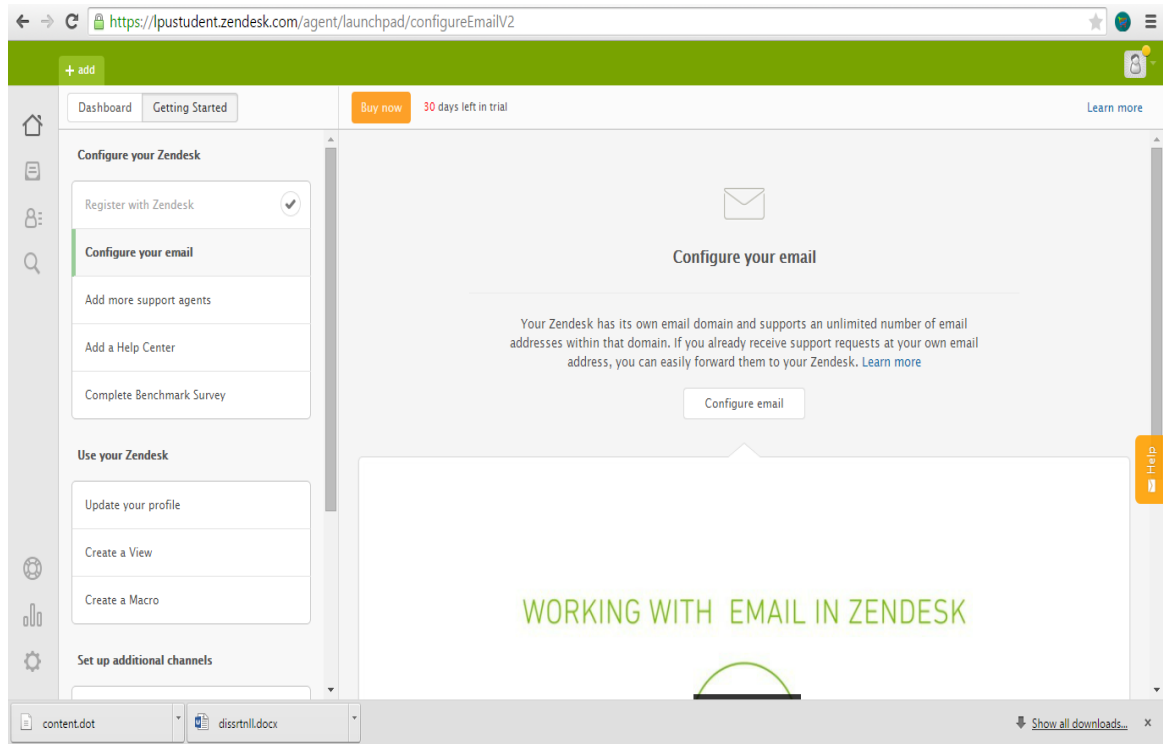


The image shows a login form for 'secingrid'. It features two input fields: the first contains the email address 'hraman.rvs@gmail.com' and the second contains a masked password '.....'. Below the password field is a checkbox labeled 'Stay signed in'. A security notice states 'Your credentials will be sent over a secure connection'. A dark 'Sign in' button is positioned to the right of the notice. At the bottom, there are two links: 'I am an Agent' and 'Forgot my password'.

Have you emailed us? [Get a password](#)

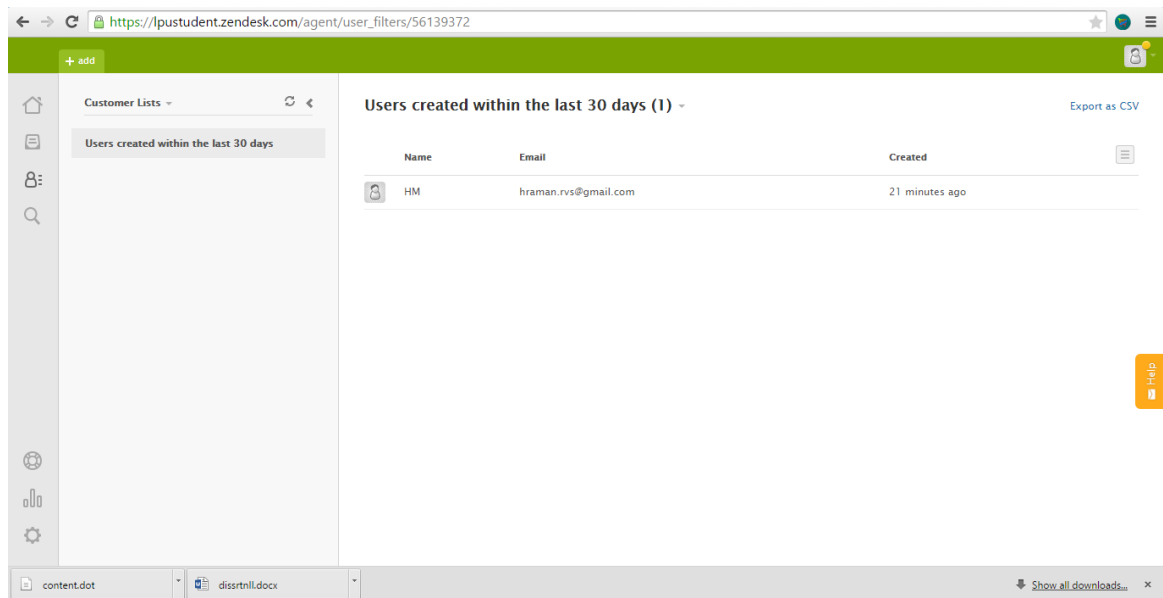
If you've communicated with our support staff through email previously, you're already registered. You probably don't have a password yet, though.

**Figure 15:** Agent login page.



**Figure 16:** Configuring the server.

We use this home page to show the Profile information and getting tokens and etc. The view function provides us jobs or tokens list and we can also see solved & unsolved tokens also. From here we can see customer list and manage them as well.



**Figure 17:** Customer list page.



+ add
🔒

**Views** ↻ ←

- Your unsolved tickets 13
- Unassigned tickets 0
- All unsolved tickets 14
- Recently updated tickets 1
- New tickets in your groups 0
- Pending tickets 0
- Recently solved tickets 0
- Unsolved tickets in your groups 14
- Suspended tickets 0

More >

## Your unsolved tickets

13 tickets Play ▼

	Subject	Requester	Requested	Type	Priority
Status: Open					
🗑	<span style="color: red;">o</span> OWfhOIClm4j3Ug5dxaotmw==	himansu	5 minutes ago	Ticket	Normal
🗑	<span style="color: red;">o</span> GRHW7ClePjMaY8TOhog8w==	himansu	Yesterday 04:55	Ticket	Normal
🗑	<span style="color: red;">o</span> 19yl4jrCuL1UathFPYSbuw==	himansu	Yesterday 04:44	Ticket	Normal
🗑	<span style="color: red;">o</span> 19yl4jrCuL1UathFPYSbuw==	himansu	Yesterday 04:44	Ticket	Normal
🗑	<span style="color: red;">o</span> ody5rBj0wu0=	himansu	Yesterday 04:38	Ticket	Normal
🗑	<span style="color: red;">o</span> rQzq8QQ4jH=	himansu	Yesterday 04:31	Ticket	Normal
🗑	<span style="color: red;">o</span> mXMCGRHMK1UathFPYSbuw==	himansu	Yesterday 04:27	Ticket	Normal
🗑	<span style="color: red;">o</span> BroLwfaFzYs=	himansu	Yesterday 04:19	Ticket	Normal
🗑	<span style="color: red;">o</span> CSqsIbcOIL3tb3aj5iVhiw==	himansu	Yesterday 03:05	Ticket	Normal
🗑	<span style="color: red;">o</span> y57wl7R112HIs0ioOFoqpQ==	himansu	Yesterday 03:04	Ticket	Normal
🗑	<span style="color: red;">o</span> 0yyFvgdyOF7vagz0r3ANew==	himansu	Yesterday 02:59	Ticket	Normal
🗑	<span style="color: red;">o</span> GjXbUYwvNm4=	himansu	Yesterday 02:48	Ticket	Normal
🗑	<span style="color: red;">o</span> CgAT868taOE+DW2iup0gHg==	himansu	Yesterday 02:40	Ticket	Normal

Help

**Figure 18:** Unsolved tickets.

## The Response are Decripted

### Response

subject	priority	status	type
lllllllll	normal	open	
jhg	normal	open	
final testing	normal	open	
second test	normal	open	
Third Test	normal	open	
lklsj	normal	open	
himanshu	normal	open	
shekhar	normal	open	
dhf	normal	open	
hghghghg	normal	open	
hghghghg	normal	open	
sfdhjjfg	normal	open	
testing request	normal	open	

[Learn more »](#)

**Figure 19:** Response decrypted at the server.

### CONCLUSION AND FUTURE SCOPE

---

Grids usefulness will be significant if it can be accessed by more and more users. Access to more users will utilize the resources provided by Grid. We can make it possible by using Internet infrastructure.

Finding of the study of cryptanalysis on AES is that, various methods are in progress to break AES completely, but the progress is too slow and no breakthroughs have been reported yet. For side-channel attacks various available countermeasures are present. So, we can conclude here that it is safe to use AES for encrypting communication if done carefully for secure communication in distributed environment.

The required processor speed, memory access speed, and cryptanalytic tools improvement will take time. So the deciphering of cipher in considerable and useful amount of time is difficult in near future. So in coming time we can safely use AES as the cipher for our purpose.

We can use AES for encrypting and decrypting data till the computing power like processor speed, faster memory access, and cryptanalytic studies improves drastically. So deciphering cipher to be done in considerable amount of time requires a lot of advances in technology and our knowledge of cryptanalysis. If we will apply constraints which will make side-channel attack ineffective than we can apply AES to make our system secure. We also need to keep track of advances in cryptanalytic concepts and tools regularly.

## Chapter 6

### REFERENCES

---

- [1] An Overview of Cryptanalysis Research for the Advanced Encryption Standard Alan Kaminsky<sup>1</sup>, Michael Kurdziel<sup>2</sup>, Stanisław Radziszowski<sup>1</sup> <sup>1</sup>Rochester Institute of Technology, Rochester, NY <sup>2</sup>Harris Corp., RF Communications Div., Rochester, NY ark@cs.rit.edu, mkurdzie@harris.com, [spr@cs.rit.edu](mailto:spr@cs.rit.edu)
- [2] N. Courtois, A. Klimov, J. Patarin, A. Shamir, “Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations”, EUROCRYPT, LNCS 1807, pp.392-407, Springer, 2000
- [3] B. Schneier, "Adi Shamir's Cube Attacks  
"http://www.schneier.com/blog/archives/2008/08/adi\_shamirs\_cub.html, August 19, 2008.
- [4] N. Potlapally, A. Raghunathan, S. Ravi, N. Jha., R. Lee, “Aiding Side- Channel Attacks on Cryptographic Software with Satisfiability-Based Analysis”, IEEE Transactions on VLSI Systems, 15(4), pp.465-470, April 2007.
- [5] K. Schramm, G. Leander, P. Felke, C. Paar, “A Collision-Attack on AES Combining Side Channel and Differential-Attack”, Cryptographic Hardware and Embedded Systems - CHES 2004, 6th International Workshop, Cambridge, MA, USA, 2004
- [6] O. Faurax, T. Muntean, “Security Analysis and Fault Injection Experiment on AES”, Proceedings of SAR-SSI 2007, 2007.
- [7] An Improved RSA Signature Algorithm based on Complex Numeric Operation Function. Hongwei Si, Youlin Cai, Zhimei Cheng School of mathematics and Information Science, East China Institute of Technology, Fuzhou, China.

1. AES: Advanced Encryption Standard.
2. RSA algorithm: Ron Rivest, Adi Shamir, Leonard Adleman algorithm.
3. CBC: Cipher block chaining.
4. MAC: Message authentication code.
5. SHA: Secure Hash Algorithm.